

# TBD 처리를 위한 레이더용 파티클 필터 기법 연구

## Radar Tracking Using Particle Filter for Track-Before-Detect(TBD)

권 지 훈 · 강 성 철 · 곽 노 준\*

Ji-Hoon Kwon · Seung-Chul Kang · No-Jun Kwak\*

### 요 약

본 논문은 추적-후-탐지 처리(TBD: Track Before Detect)를 위한 레이더용 파티클 필터(Particle filter)에 대해서 기술한다. TBD 기법은 강한 클러터 환경, 작은 RCS 타겟 및 스텔스 타겟 등으로 인해 타겟 탐지가 어려운 경우(낮은 SNR)에 적용하는 기술이다. 특히 파티클 필터는 재귀적 TBD(Recursive TBD) 알고리즘 구현에 적합하고, 비선형 모델을 가우시안 선형 모델로 근사화해서 추정하는 칼만 필터 대비, 상대적으로 개선된 정확도를 갖는다. 본 논문에서는 다수의 관측값(클러터 포함)들이 동시에 수신될 때, 신호강도-거리-도플러 정보를 활용하여 파티클 필터 가중치를 직접 계산 및 갱신하는 방식을 제안한다. 성능 분석을 위해 가상의 레이더 시뮬레이션 사나리오를 설정하고, 제안하는 파티클 필터를 적용하여 추적 필터의 추정오차를 분석한다.

### Abstract

This paper describes the technique for Radar Particle filter for TBD(Track Before Detect) processing. TBD technique is applied when target is difficult to detect due to low signal-to-noise ratio caused by strong clutter environments, small RCS targets and stealth targets. Particle filter is suitable for a recursive TBD algorithm and has improved estimation accuracy than Kalman filter. In this paper, we will present a new method of calculating particle weight, when observation values(including strong clutter) are received at the same time. Estimation error performance of the particle filter algorithm is analyzed by using the virtual radar observation scenario.

Key words: Radar Tracking Filter, Particle Filter, TBD, Track-Before-Detect, Stealthy Target

## I. 서 론

전통적인 레이더는 탐지-후-추적 처리(DBT: Detect Before Track) 방식을 적용한다. 특정한 임계치와의 비교를 통해서 탐지 여부를 판단한 후, 추적을 수행하는 기법으로, 일반적으로 CFAR 알고리즘과 칼만 추적 필터를 통해 구현한다.

한편, 추적-후-탐지 처리(TBD: Track Before Detect) 방

식은 열악한 클러터 환경, 작은 RCS 타겟 및 스텔스 타겟 등으로 탐지가 어려운 경우(낮은 SNR) 등에 적용한다<sup>[1]~[4]</sup>. 통상 레이더는 90 % 이상의 탐지확률과  $10^{-6}$  이하의 오경보확률을 유지하려고 위해 높은 수준의 임계치를 요구한다. 설정된 높은 수준의 임계치는 작은 SNR을 가진 표적 또한 제거해 버린다. 따라서 기존 DBT 방식에서는 작은 RCS를 가진 표적 등이 탐지되지 않기 때문에, 추적 처리에 제한을 받는다<sup>[4],[10]</sup>.

「이 논문은 2015년도 한화탈레스(주)의 재원을 지원 받아 수행된 연구임.」

한화탈레스 레이더 · 전자전연구소(Radar · EW R&D Center, Hanwha THALES)

\*서울대학교 융합과학기술대학원(Graduate School of Convergence Science and Technology, Seoul National University)

· Manuscript received January 26, 2016 ; Revised February 26, 2016 ; Accepted March 7, 2016. (ID No. 20160126-011)

· Corresponding Author: No-Jun Kwak (e-mail: nojunk@snu.ac.kr)

TBD 처리 기법은 임계값에 대한 필요성을 제거하고, 낮은 SNR 상황에서도 타겟을 추적/탐지하는 데 유용하다<sup>[4],[10]</sup>. TBD 알고리즘은 대표적으로 동적 계획법(dynamic programming) 기법과 파티클 필터(particle filter) 기법이 널리 적용되는 것으로 보고되었다<sup>[5],[8],[10]</sup>. 동적 계획법은 움직이는 물체를 검출하는 기법으로 Viterbi 알고리즘을 통해서 구현한다. 그러나 동적계획법 기반의 알고리즘은 추적의 트랙이 분리되어 추적 성능이 떨어지는 것으로 보고되었고<sup>[5]~[7]</sup>, 일정량의 스캔 데이터를 모아서 처리하는 기법(batch method)으로 알려졌다<sup>[8],[10]</sup>.

이러한 문제의 대안으로 파티클 필터(particle filter)가 널리 연구되고 있다<sup>[5],[8]</sup>. 파티클 필터(particle filter)는 재귀적 방식으로 경로를 연속적으로 추정하기 때문에, 재귀적 TBD(Recursive TBD) 알고리즘 구현에 적합하다<sup>[4],[8],[10]</sup>. 파티클 필터는 가우시안 선형 모델로 근사화하는 칼만 필터(Kalman filter) 대비 비선형모델에 대한 추정 오차를 줄이기 때문에, 개선된 정확도를 갖는다. 특히 작은 RCS를 갖는 표적 등의 경우, 비선형적인 특성을 갖기 때문에 칼만 필터보다 파티클 필터가 적합하다.

본 논문은 TBD 처리를 위한 레이더용 파티클 기법에 대해서 기술한다. 2장에서는 파티클 필터의 이론적인 부분을 기술하고, 특히 복수의 관측값(클러터 포함)이 동시에 수신될 때, 가중치를 계산하는 방식을 제안한다. 3장은 가상의 시나리오를 설정하고, 고안된 알고리즘을 적용하여 추적처리를 수행한다. 4장은 연구결과 및 향후방향에 대해서 기술한다.

## II. 파티클 필터

### 2-1 시스템 상태방정식(System State Model)

시스템 상태방정식을 정의하기 위해 타겟의 상태를 다음 식 (1)과 같이 정의한다. 이때, 변수는 각각 x 좌표, y좌표, z 좌표, x방향 속도, y방향 속도, z방향 속도를 뜻하며, k는 인덱스이다.

$$s_k = [x_k, y_k, z_k, \dot{x}_k, \dot{y}_k, \dot{z}_k]^T \quad (1)$$

정의된 상태에 대한 시스템 상태방정식을 식 (2)와 같이 정의하고, 이때 A 및 W 행렬을 식 (3)에 보인다. A 행

렬은 위치와 속도 간의 관계를 나타내는 행렬이며, W는 가속도 오차에 의해 발생하는 각각 위치 및 속도에 대한 오차를 의미한다.

$$s_k = A \cdot s_{k-1} + W \quad (2)$$

$$A = \begin{bmatrix} 1 & 0 & 0 & T & 0 & 0 \\ 0 & 1 & 0 & 0 & T & 0 \\ 0 & 0 & 1 & 0 & 0 & T \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}, W = \begin{bmatrix} 0.5(\sigma_{a,x}^2)T^2 \\ 0.5(\sigma_{a,y}^2)T^2 \\ 0.5(\sigma_{a,z}^2)T^2 \\ (\sigma_{a,x}^2)T \\ (\sigma_{a,y}^2)T \\ (\sigma_{a,z}^2)T \end{bmatrix} \quad (3)$$

식 (3)에서 T는 상태가 갱신되는 시간을 의미하고, W 행렬의  $\sigma^2$ 는 각각 해당하는 축 방향의 가속도 오차 분산을 의미한다.

### 2-2 관측모델(Measurement Model)

관측자 입장에서 시스템의 상태를 측정(관측, 측정, 센싱 등)한 값에 대한 해석을 관측모델이라 하며, 식 (4)와 같이 정의한다. 함수  $h(\cdot)$ 는 상태값  $s_k$ , 잡음  $n_k$ 에 대한 관측값  $z_k$ 와의 관계이다<sup>[11]</sup>.

$$z_k = h(s_k, n_k) \quad (4)$$

파티클 필터의 핵심은 상태공간에서 N개의 파티클들로 식 (6)의 확률을 최적화하는 것으로, 이는 과거부터 현재까지 수집한 관측값에 대한 조건부인 사후확률분포를 통해 근사화한다<sup>[11]</sup>.

$$Z_k = \{z_0, \dots, z_k\} \quad (5)$$

$$p(s_k | Z_k) \quad (6)$$

이 사후확률분포는 N개의 파티클들로 표현하는데, 이를 식 (7)에 보인다<sup>[11]</sup>.

$$\{s_k^i\}_{i=1 \dots N} \quad (7)$$

칼만 필터는 가우시안 분포로 근사화하여 추적처리를 수행하기 때문에 실제 확률 모델과의 차이로 인한 추정 오차가 발생한다. 하지만, 파티클 필터는 N개의 파티클을 통해서 확률분포를 근사화하고, 새로운 관측값을 통해 지

속적으로 파티클의 분포(확률분포)를 갱신하기 때문에, 가우시안 분포가 아닌 비선형 확률 분포 문제(특히 작은 RCS를 가진 타겟) 등에 대한 추정 오차를 줄인다.

$N$ 개의 파티클은 각각 가중치(중요도)를 갖고 있으며, 가중치와 파티클의 위치를 통해 추정 상태를 식 (8)과 같이 계산한다.

$$\hat{s}_k \approx \frac{1}{N} \sum_{i=1}^N s_k^i \quad (8)$$

관측모델에서 중요하게 계산되어야 할 것은 파티클의 가중치를 계산하여 확률분포에 반영하는 것이다. 본 논문에서는 CFAR의 임계치를 낮췄을 때, 다수의 관측값들(실제 관측값 + 클러터)이 동시 시각에 수신되는 경우를 가정하고, 총 3가지 방식에 대해서 비교 분석한다. 첫 번째 방식은 다수의 관측값들이 가진 신호강도-거리-도플러 정보를 활용하여 한 개의 가중치 평균 좌표값을 구하고, 확장칼만필터(EKF)를 통해 추정값을 구하는 방법이다. 두 번째 방식은 첫 번째 방식과 동일하지만, 확장칼만필터 대신 파티클 필터를 사용한다. 세 번째 방법은 본 논문에서 제안하는 방식으로, 다수의 관측값들의 가진 정보를 사용해서, 파티클 필터의 가중치를 직접 계산하는 방법이다. 세 가지 방법에 대한 특징을 표 1에 정리한다. 아울러, 본 논문에서 제안하는 방식인 세 번째 방법에 대한 개념을 그림 1에 보인다.

방식 (A)와 (B)에서 사용하는 가중치 평균 좌표값은 수식 (9), (10) 및 (11)을 통해 계산한다. 이때,  $PD$ 는 탐지확률로 관측값의 SNR과 오경보확률인  $P_{fa}$ 을 통해 계산한다.  $k$  번째 인덱스에서 다수의 관측값이  $m$ 개 있을 때,  $m$ 번째 관측값에 대한 탐지확률  $PD_m$ 을 식 (9),  $m$ 번째 관측값에 대한 가중치  $w_m$ 을 식 (10)을 통해서 구한다. 이때  $Q(\cdot)$  함수는 Marcum's Q-function이다. 거리  $r_m$ 은 이전 추정 좌표와 해당 관측값과의 거리 차이를 통해서 계산하고, 속도 정보  $v_m$ 는 도플러 값을 통해 계산한다. 각각의 관측값에 대한 가중치  $w_m$ 으로부터 정규화된 가중치  $w = \{w_1, w_2, \dots, w_m\}$ 을 구하고, 이를 반영하여 가중치 평균 좌표값을 식 (11)을 통해서 구한다.

$$PD_m = Q(\sqrt{2(SNR_m)}, \sqrt{-2\log_e P_{fa}}) \quad (9)$$

표 1. 본 논문에서 적용한 추적기법의 특징  
Table 1. Characteristics of tracking methods applied in this paper.

| 구분  | 입력 관측값                   | 적용필터   |
|-----|--------------------------|--------|
| (A) | 가중치 평균좌표값                | 확장칼만필터 |
| (B) | 가중치 평균좌표값                | 파티클 필터 |
| (C) | 다수 관측값으로부터 파티클 가중치 직접 계산 | 파티클 필터 |

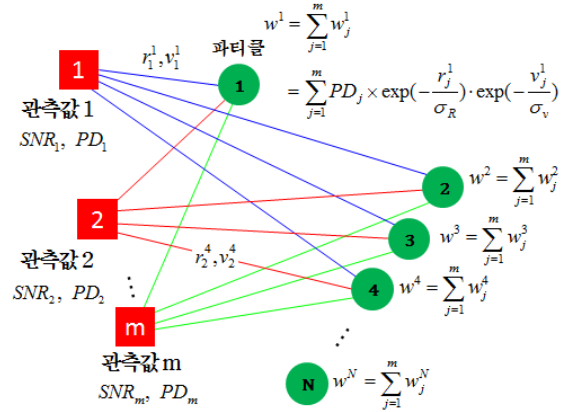


그림 1. 본 논문에서 제안하는 가중치 계산 방법  
Fig. 1. The proposed weight calculation method.

$$w_m = PD_m \cdot e\left(-\frac{r_m}{\sigma_r}\right) \cdot e\left(-\frac{v_m}{\sigma_v}\right) \quad (10)$$

$$\begin{aligned} x \text{ positon} &= \frac{1}{m} \sum_{j=1}^m w_j x_j \\ y \text{ positon} &= \frac{1}{m} \sum_{j=1}^m w_j y_j \\ z \text{ positon} &= \frac{1}{m} \sum_{j=1}^m w_j z_j \end{aligned} \quad (11)$$

특히 본 논문에서 제안하는 ‘다수 관측값을 통한 파티클 가중치 직접 계산’ 방법에 대한 상세 설명은 다음과 같다. 앞서 제시한 그림 1에서 사각형 점은 해당 시각에 수신된 관측값들이고, 둥근 점은 파티클의 분포이다.  $k$  번째 인덱스에서  $m$ 개의 관측값들을 통해  $N$ 개의 파티클 가중치를 다음 식 (12), (13), (14) 및 (15)를 통해 계산한다.

$$PD_m = Q(\sqrt{2(SNR_m)}, \sqrt{-2\log_e P_{fa}}) \quad (12)$$

$$w_m^i = PD_m \cdot e^{-\frac{r_m^i}{\sigma_r}} \cdot e^{-\frac{v_m^i}{\sigma_v}} \quad (13)$$

$$\overline{w_s} = \left[ \sum_{j=1}^m w_j^1 \quad \sum_{j=1}^m w_j^2 \quad \dots \quad \sum_{j=1}^m w_j^N \right] \quad (14)$$

$$w = \overline{w_s} \div \text{sum}(\overline{w_s}) = \{w^1 \dots w^N\} \quad (15)$$

m번째 관측값과 i번째 파티클 간에 가중치를 식 (13)과 같이 계산한다. m개의 관측값에 대한 파티클 N개의 가중치 합을 식 (14)와 같이 계산하고, 정규화된 가중치를 식 (15)를 통해 구한다.

2-3 리샘플링(Re-sampling)

파티클 필터는 리샘플링(Re-sampling) 과정을 통해서 중요도가 높은 파티클을 균등 가중치로 분할하고, 중요도가 낮은 파티클을 제거한다. 리샘플링을 수행하기 위해서 유효파티클수  $N_{eff}$ 를 정의하고, 이 값이  $N_{th}$  이하일 때 리샘플링을 수행한다. 이를 식 (16)에 보인다. 델타함수( $\delta$ )를 사용하여 파티클의 위치와 각각의 가중치를 가진 형태의 확률 분포를 식 (17)에 보인다. 리샘플링 수행 후, 가중치가 균등하게 분할된 형태를 식 (18)에 보인다.

$$N_{eff} = 1 \div \left( \sum_{i=1}^N (w_k^i) \right) \quad (16)$$

$$p(s) \approx \sum_{i=1}^N w_i \delta(s - s_k^i) \quad (17)$$

$$\hat{p}(s) \approx \sum_{i=1}^N \frac{1}{N} \delta(s - \tilde{s}_k^i) \quad (18)$$

식 (18)를 통해 확률분포를 가중치가 균일한 N개의 파티클로 표현하며, 리샘플링 후의 파티클들의 상태값을 식 (8)에 대입하여 위치를 추정한다. 다음 시각 인덱스에서 각각의 파티클은 상태방정식 식 (2)를 통해서 상태값이 갱신되는 방식으로 재귀적인 처리를 수행한다.

III. 시뮬레이션

3-1 레이더 시뮬레이션 시나리오 설정

해석을 위해 저피탐 비행체의 실 이동 경로를 그림 2와

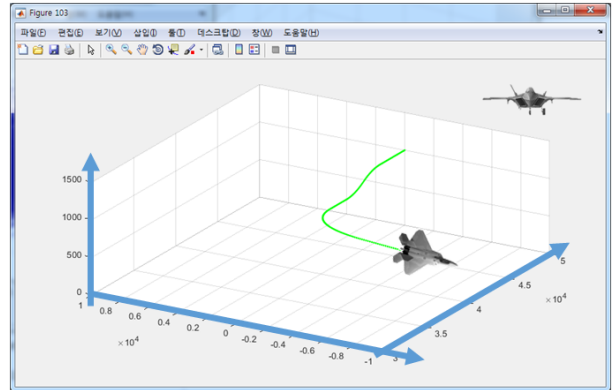


그림 2. 저피탐 타겟 이동 경로 설정  
Fig. 2. Moving path scenario of low-intercept target (low SNR & low RCS target).

표 2. 시뮬레이션을 위한 레이더 시스템 파라미터  
Table 2. Radar system parameter for simulation.

| System parameter     | 심볼    | 값                |               |
|----------------------|-------|------------------|---------------|
| Operating frequency. | $f$   | L-band           |               |
| Transmit peak power  | $P_t$ | -                |               |
| Pulse width          | $T$   | -                |               |
| N(# of pulse)        | $N$   | 100              |               |
| Tx antenna gain      | $G_t$ | 35 dBi           |               |
| Rx antenna gain      | $G_r$ | 35 dBi           |               |
| System noise figure  | $F$   | 10 dB            |               |
| RCS                  | Front | $\sigma_{front}$ | Ref. dBsm     |
|                      | Top   | $\sigma_{top}$   | (Ref.+5) dBsm |
|                      | Side  | $\sigma_{side}$  | (Ref.+3) dBsm |

같이 설정한다. 타겟은 x축으로 레이더와 가까운 방향으로 근접하다가 z축으로 일정부분 고도를 하강한 후에, y축으로 급선회하여 빠져나간다.

분석을 위해 표 2와 같이 레이더 시스템 파라미터를 설정한다. RCS가  $\sigma_{front}$ ,  $\sigma_{top}$ ,  $\sigma_{side}$  순으로 변하고, RCS 값의 편차가 있다고 가정한 후에, 레이더방정식을 저피탐 타겟 이동 경로에 따른 SNR을 계산한다. 이를 그림 3의 붉은점으로 보인다.

이동 경로에 복잡한 클러터를 생성 반영하여 시뮬레이션 시나리오를 생성한다. 이를 그림 4에 보인다.

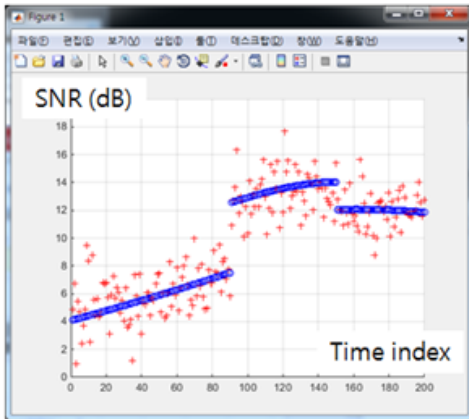


그림 3. 이동 경로에 따른 SNR  
Fig. 3. SNR according to the simulation scenario.

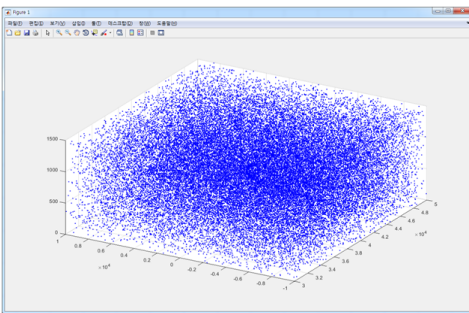


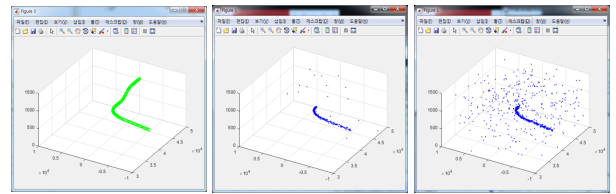
그림 4. 레이더 클러터를 포함한 시뮬레이션 시나리오  
Fig. 4. Simulation scenario including radar clutter.

### 3-2 임계치 설정에 따른 관측값

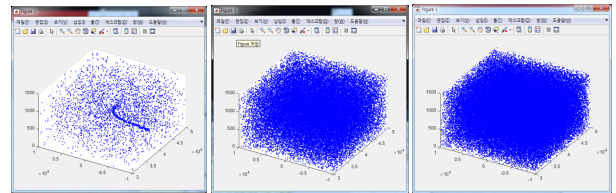
임계치 설정에 따른 관측경로 및 클러터의 변화를 그림 5에 보인다. 임계치가 높을수록 클러터를 효과적으로 제거할 수 있으나, 표적의 SNR이 낮은 구간에서 관측값이 존재하지 않는다. 반대로, 임계치를 낮추면 복잡한 클러터까지 수신되기 때문에, 선형화된 칼만필터 적용에 한계가 있다.

### 3-3 관측값 처리 범위 설정

SNR 임계치를 낮추면 특정한 시기에 유입되는 관측값이 매우 많고, 특히 전체 공간상에 관련 없는 클러터 값들이 존재한다. 따라서 추정된 값을 기준으로 반경  $R$  이내의 값만을 처리토록 구현한다. 이때 반경  $R$ 은 타겟의 이동속도를 고려하여 설정한다. 이를 그림 6에 보인다.



(a) 실제 이동경로 (a) Real path  
(b) SNR=12 dB 이상 (b) Measurement path (SNR Threshold level=12 dB)  
(c) SNR=9 dB 이상 (c) Measurement path (SNR Threshold level=9 dB)



(d) SNR=7 dB 이상 (d) Measurement path (SNR Threshold level=7 dB)  
(e) SNR=5 dB 이상 (e) Measurement path (SNR Threshold level=5 dB)  
(f) SNR=3 dB 이상 (f) Measurement path (SNR Threshold level=3 dB)

그림 5. SNR 임계치에 따른 관측경로 및 클러터  
Fig. 5. Measurement path with clutter according to SNR threshold level.

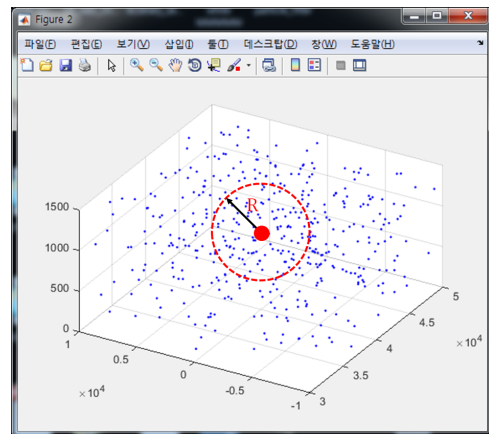
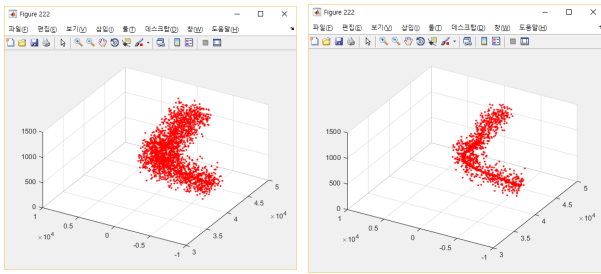


그림 6. 효율적인 처리를 위한 관측값 처리 범위 제한  
Fig. 6. Constraints of processing measurement values for improving efficiency.

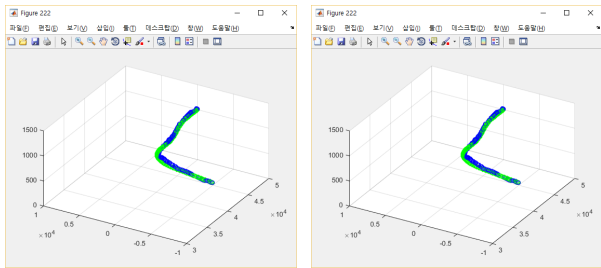
### 3-4 시뮬레이션

그림 4에서 보인 시뮬레이션 시나리오를 가지고, 임계치 SNR 고정 5 dB 이하를 제거하고, 관측값 처리 범위를 각각 2.5 km, 1 km로 제한한 후에 수신된 관측값(클러터 포함)을 그림 7에 보인다.



(a) 제한범위 2.5 km (a) Limitation R=2.5 km  
 (b) 제한범위 1 km (b) Limitation R=1 km

그림 7. 관측값 처리 범위 제한 후, 관측값  
 Fig. 7. Measurement values after limiting processing area.



(a) 제한범위 2.5 km (a) Limitation R=2.5 km  
 (b) 제한범위 1 km (b) Limitation R=1 km

그림 8. 실제경로(녹색)와 추정경로(파란색) : 파티클 수 100  
 Fig. 8. Real path vs estimated path : Particle number 100.

그림 7에서 보인 두 가지 경우에 대한 파티클 필터 추정 결과에 대한 결과를 그림 8에 보인다. 두 경우의 추정오차의 차이는 미미한 수준이지만, 관측값의 수가 많은 제한 범위의 2.5 km가 상대적으로 많은 연산량을 요구한다. 추정오차에 대해서는 3-4장에 기술한다.

파티클 필터는 파티클 수가 많을수록 정확도가 개선된다. 하지만 증가한 파티클 수에 비례적으로 연산량이 증가하는 문제점을 갖고 있다. 그림 9는 파티클 수를 600으로 증가시켰을 때의 추정경로와 실제경로와의 차이를 보인다. 입력되는 관측값이 그림 7에서 보여지는 것과 같이 복잡함에도 불구하고, 실제 경로를 잘 추정하는 것을 확인할 수 있다.

### 3-5 추정 성능 분석

추정 성능은 RMSE(Root Mean Square Error)를 통해서

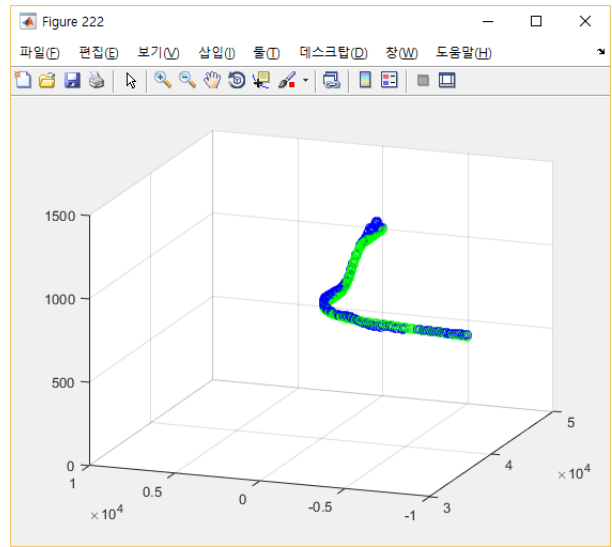


그림 9. 실제경로(녹색)와 추정경로(파란색) : 파티클 수 600  
 Fig. 9. Real path vs estimated path : particle number 600.

분석한다. 관측오차 및 추정오차를 각각 식 (19)와 식 (20)을 통해서 구한다. 식 (19)는 특정 시각 인덱스  $k$ 에서 관측값이  $n$ 개일 때의 계산되는 관측오차이다. 식 (20)은 특정 시각 인덱스  $k$ 에서의 추정 오차를 나타낸다. 여기서  $M_{error}$ 는 관측오차,  $M_{pos}$ 는 관측값의 위치,  $R_{pos}$ 는 실제경로의 위치,  $E_{error}$ 는 추정오차,  $E_{pos}$ 는 추정값의 위치를 나타낸다.

$$M_{error}(k) = \frac{1}{n} \sum_{i=1}^n \| M_{pos}(k)_i - R_{pos} \|_2 \quad (19)$$

$$E_{error}(k) = \| M_{pos}(k) - R_{pos} \|_2 \quad (20)$$

그림 10은 관측오차와 추정오차(파티클 수 600)를 보인다. 이 때,  $R$ 은 관측값 처리 범위 제한을 나타낸다. 복잡한 클러스터 및 낮은 SNR 환경에서도 파티클 필터를 통한 추정오차 성능이 우수함을 알 수 있다.

적정 파티클 수를 분석하기 위해 평균 추정오차  $E_{error,avg}$ 를 식 (21)과 같이 정의하고, 파티클 수에 따른 추정오차와 평균추정오차를 각각 그림 11과 그림 12에 보인다. 각각의 값은 20회의 시뮬레이션을 통한 평균값이다.

$$E_{error,avg} = \frac{1}{k} \sum_{idx=1}^k E_{error}(k) \quad (21)$$



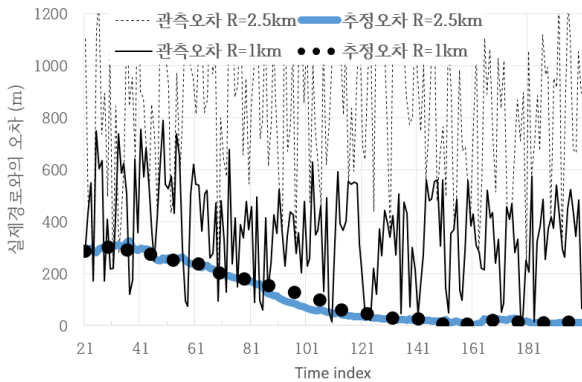


그림 10. 관측오차와 추정오차(파티클 수: 600)  
Fig. 10. Measurement error vs estimation error.

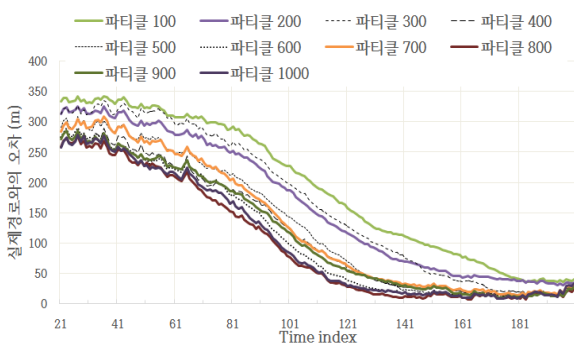


그림 11. 파티클 수에 따른 추정오차  
Fig. 11. Estimation error according to the number of particles.

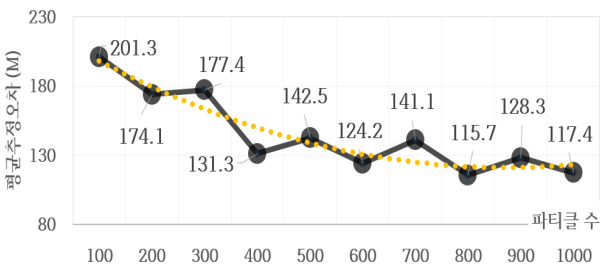


그림 12. 파티클 수에 따른 평균추정오차  
Fig. 12. Average estimation error according to the number of particles.

그림 12의 결과를 통해 파티클 수가 일정 수준에 도달하면 추정 오차가 수렴함을 알 수 있다. 본 연구에서는 대략 600개 수준이 설정된 시나리오에서의 오차가 일정 수

준 이하로 수렴하는 유효 개수로 분석된다.

### 3-6 기존 방식(칼만 및 파티클 필터)들과의 성능 비교

표 1에 본 논문에서 비교 분석할 3가지 방식에 대해서 기술하였다. 동일한 SNR 및 처리 범위 제한 조건을 기반으로 3가지 방식에 대해 시뮬레이션을 수행하였고, 추정 오차 및 평균추정 오차 결과를 각각 그림 13과 표 3에 보인다.

결론적으로 본 논문에서 제안하는 방식인 (C) 방식이 가장 우수한 성능을 보였으며, 제안 방식은 SNR이 낮은 상황뿐만 아니라, 높은 상황에서도 실제 경로와의 오차가 가장 작았다. 한편, (A) 방식인 확장칼만필터 기반 추정 알고리즘은 3가지 방식 중에 추정 오차 성능이 가장 좋지

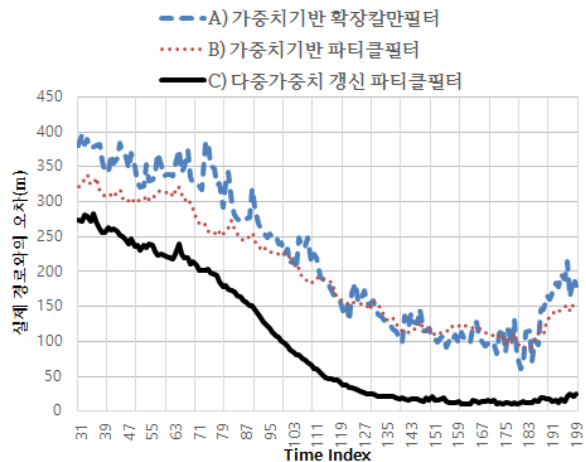


그림 13. 알고리즘 방식에 따른 추정오차 비교  
Fig. 13. Estimation error according to the tracking algorithm.

표 3. 알고리즘 방식에 따른 평균추정오차 비교  
Table 3. Average estimation error according to the tracking algorithm.

| 구분  | 특징              | 평균추정오차 (k=1~100) | 평균추정오차 (전구간) |
|-----|-----------------|------------------|--------------|
| (A) | 확장칼만필터          | 332 m            | 236 m        |
| (B) | 파티클 필터          | 291 m            | 214 m        |
| (C) | 파티클 필터 (본 논문제안) | 221 m            | 124 m        |

※ k는 time index를 의미한다.

않았다. 특히 SNR이 낮은 상황에서 파티클기반 알고리즘인 (C)방식 대비 2배 이상의 추정 오차를 보였다. 또한, 확장칼만필터의 경우, 파티클 필터를 적용한 알고리즘 대비 추정오차 변동 폭 상대적으로 컸다.

#### IV. 결 론

본 연구에서는 다수의 관측값들(실제관측값 + 클러터)이 동시 시각에 수신될 때, 신호 강도-거리-도플러 정보를 활용하여 파티클 가중치를 직접 계산 및 갱신하는 방법을 제안하였고, 이를 적용한 TBD용 파티클 필터 기법을 구현하였다. 구현한 알고리즘으로 타겟을 추적한 결과, 가중치 평균 좌표값을 사용하는 확장칼만필터 및 파티클 필터 TBD 알고리즘의 추정오차 대비 2배 이상의 향상된 성능을 갖는 것으로 분석하였다.

#### References

[1] D. J. Salmond, H. Birch, "A particle filter for track-before-detect", *Proceedings of the American Control Conference*, vol. 5, pp. 3755-3760, 2001.

[2] Bocquel, Mélanie, Hans Driessen, and Arun Bagchi, "Multitarget particle filter addressing ambiguous radar data in TBD", *Radar Conference(RADAR)*, pp. 575-580, 2012.

[3] Zhaoping Wu, Tao Su, "Radar target detect using particle filter", *Radar Conference 2010 IEEE*, pp. 955-958, May 2010.

[4] Y. Boers, J. N. Driessen, "Multitarget particle filter track

before detect application", *IEE Proceedings-Radar, Sonar and Navigation*, pp. 351-357. Dec. 2004.

[5] S. Tugac, M. Efe, "Radar target detection using hidden markov models", *Progress in Electromagnetics Research B*, vol. 44, 241-259, 2012.

[6] L. A. Johnston, V. Krishnamurthy, "Performance analysis of a dynamic programming track before detect algorithm", *IEEE Transaction on Aerospace and Electronic Systems*, vol. 38, pp. 228-242, 2002.

[7] Shane M. Tonissen, Robin J. Evans, "Performance of dynamic programming techniques for track-before-detect", *Aerospace and Electronic Systems, IEEE Transactions on 32.4*, pp. 1440-1451, 1996.

[8] Mark G. Rutten, Neil J. Gordon, and Simon Maskell, "Recursive track-before-detect with target amplitude fluctuations", *IEE Proceedings-Radar, Sonar and Navigation*, pp. 345-352, 2005.

[9] Mark G. Rutten, Branko Ristic, and Neil J. Gordon, "A comparison of particle filters for recursive track-before-detect", *Information Fusion, 2005 8th International Conference on*, vol. 1, pp. 169-175, 2005.

[10] Biruk K. Habtemariam, Ratnasingham Tharmarasa, and Thia Kirubarajan, "PHD filter based track-before-detect for MIMO radars", *Signal Processing*, pp. 667-678, 2012.

[11] Y. Boers, J. N. Driessen, "Particle filter based detection for tracking", *American Control Conference*, vol. 6, pp. 4393-4379, 2001.

#### 권 지 훈



2004년 8월: 충남대학교 정보통신공학부 전기전자전파 (공학사)  
 2007년 2월: 광주과학기술원 기전공학과 (공학석사)  
 2015년~현재: 서울대학교 융합과학기술 대학원 (지능형융합시스템) 박사과정  
 2007년~현재: 한화탈레스 레이더 전자전

체계그룹 전문연구원

[주 관심분야] 레이더시스템, 신호처리, 머신러닝 및 패턴인식 등

#### 강 성 철



1999년 2월: 명지대학교 전자공학과 (공학사)  
 2001년 2월: 명지대학교 전자공학과 (공학석사)  
 2007년 3월~현재: 한화탈레스 레이더 전자전 체계그룹 수석연구원  
 [주 관심분야] 레이더시스템, 레이더 신호

/데이터 처리 등



곽 노 준



1997년 2월: 서울대학교 전기컴퓨터공학부 (공학사)

1999년 2월: 서울대학교 전기컴퓨터공학부 (공학석사)

2003년 2월: 서울대학교 전기컴퓨터공학부 (공학박사)

2003년 3월~2006년 8월: 삼성전자통신연

구소 책임연구원

2006년 9월~2007년 2월: 서울대학교 정보기술사업단 BK조교수

2007년 3월~2013년 8월: 아주대학교 전자공학부 조/부교수

2013년 9월~현재: 서울대학교 융합과학기술대학원 부교수

[주 관심분야] 패턴인식, 머신러닝, 컴퓨터비전, 통계추정